先上结论：jump\_to\_APP()函数内部如果有printf，则需要\_\_attribute\_\_((noreturn))对自定义函数类型进行声明，没有printf则一切正常。

以下是我实操的过程：

关于stm32的bootloader跳转到应用程序的应用，网上大部分是基于keil的，大致流程是

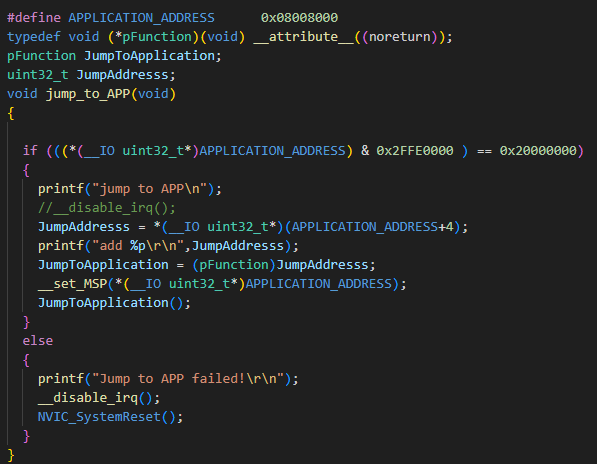
1. 在bootloader工程中实现jump\_to\_APP函数；
2. 在app工程中修改程序起始地址和中断向量表偏移；

我用gcc编译，按照以上操作不成功，尝试关闭中断也不行，最后发现给自定义的函数类型添加\_\_attribute\_\_((noreturn))声明可以了。



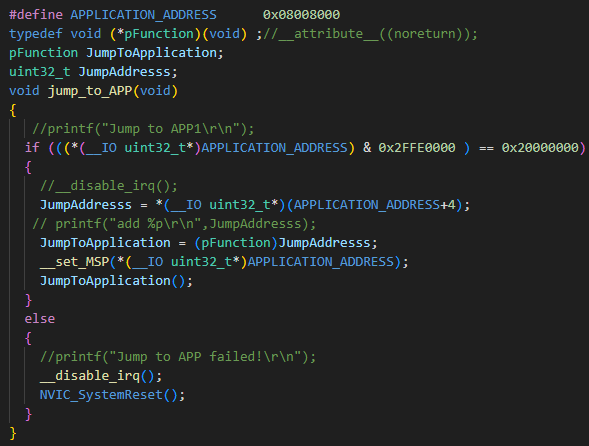
也不需要开关中断什么的，当然在bootloader工程中除了systick、usart1、gpio，其他外设都没开，这里并不能说明中断没有影响，只是我没遇到，你如果遇到奇怪的问题还是要考虑中断的可能性。

\_\_attribute\_\_((noreturn))这个声明告诉编译器这个函数不需要返回，会对程序做一定的优化。



接下来我突然灵机一动，把函数里边的printf都注释掉，然后去掉\_\_attribute\_\_((noreturn))发现程序也正常了，说明是printf影响了程序的正常运行。具体怎么影响的不清楚，有可能是调用栈的问题，也有可能是usart这个外设的原因。

这个printf真是太坑了，对于嵌入式开发，有它和没它完全是不一样的，千万不要觉得它就是打印log对程序逻辑没有影响，事实上影响大着哩，能不用尽可能不用。



最后，你开发过程中被printf坑过吗？